Sandia National Laboratories

Exceptional service in the national interest

ME469: Advection Stabilization:
Achieved through upwind and limiters

Stefan P. Domino[1,2]

[1] Computational Thermal and Fluid Mechanics, Sandia National Laboratories

[2] Institute for Computational and Mathematical Engineering, Stanford

SAND2018-4536 PE

U.S. DEPARTMENT OF ENERGY

NNSA

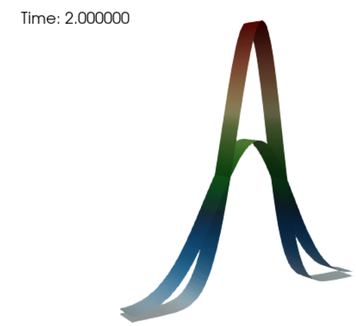
Dissipation and Dispersion Error: Review

Recall, in our modified one-dimensional advection of a passive scalar, we had two types of errors that manifested depending on the underlying numerical approach:

- Dissipative-like error:

$$\frac{\partial \phi}{\partial t} + v\frac{\partial \phi}{\partial x} - \alpha\frac{\partial^2 \phi}{\partial x^2} = 0$$

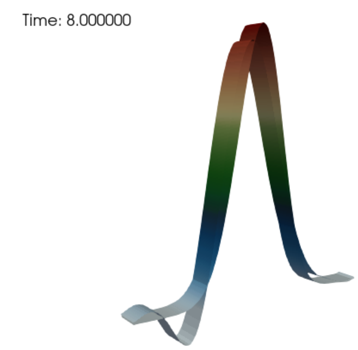
$$\phi(x,t) = e^{(-ivk-\alpha k^2)t}e^{ikx} = e^{ik(x-vt)}e^{-\alpha k^2 t}$$



- Dispersion-like error:

$$\frac{\partial \phi}{\partial t} + v\frac{\partial \phi}{\partial x} + \beta\frac{\partial^3 \phi}{\partial x^3} = 0$$

$$\phi(x,t) = e^{(-ivk+\beta i k^3)t}e^{ikx} = e^{ik[x-(v-\beta k^2)t]} = e^{ik(x-wt)}$$



Hybrid-Based Blending

For our general temporal advection/diffusion/source equation, can we define an improved, automatic blended approach between central and upwind?

$$\frac{\partial \rho \phi}{\partial t} + \frac{\partial \rho u_j \phi}{\partial x_j} - \frac{\partial}{\partial x_j}\left(\frac{\mu}{Sc}\frac{\partial \phi}{\partial x_j}\right) = S^{\phi}$$

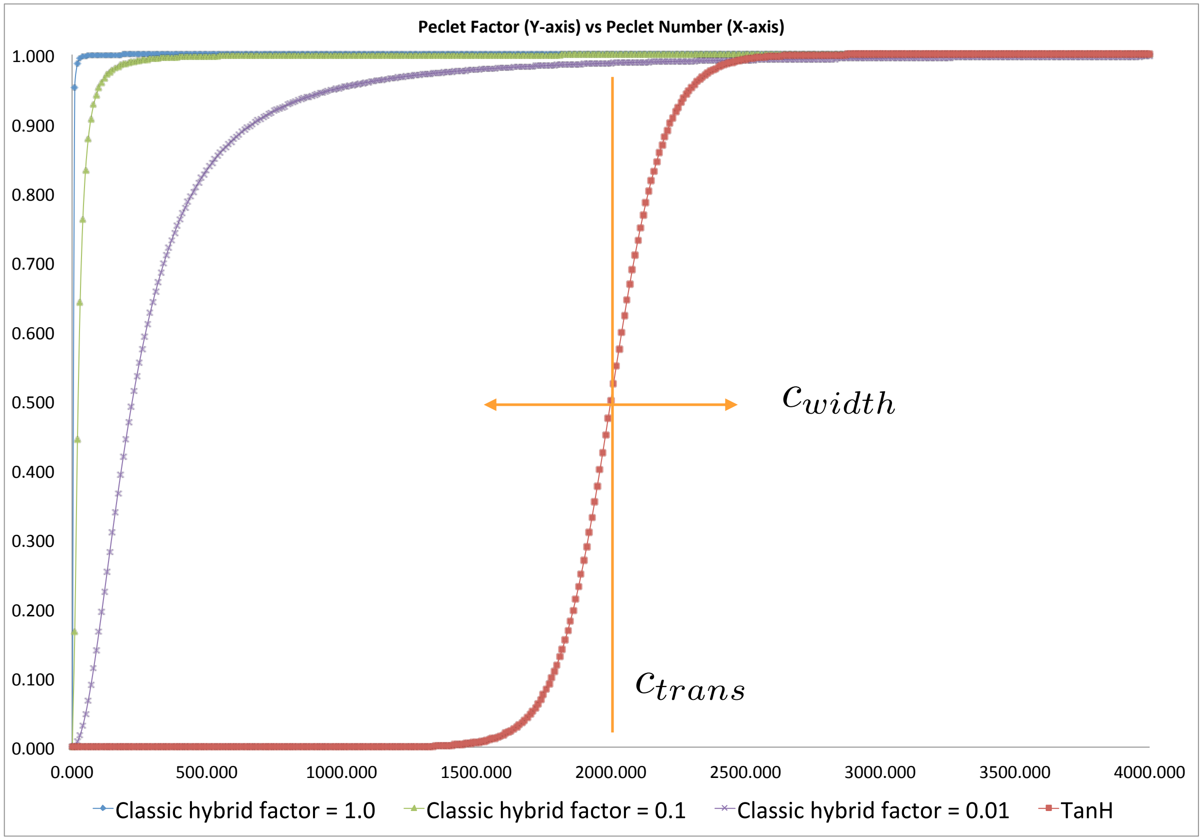
Recall, the advection operator was: $\int w \frac{\partial \rho u_j \phi}{\partial x_j} dV \approx \sum_{ip} \left(\rho u_j\right)_{ip} \phi_{ip} n_j dS \approx \sum_{ip} \dot{m}_{ip} \phi_{ip}$

With (for central, or Galerkin): $\phi_{ip}^{CDS} = \sum_n N_n^{ip} \phi_n$

And (for first-order upwind): $\phi_{ip}^{FOU} = \frac{\dot{m} + |\dot{m}|}{2}\phi_L + \frac{\dot{m} - |\dot{m}|}{2}\phi_R$

We can define a blended operator as well: $\phi_{ip} = \eta \phi^{FOU} + \left(1 - \eta\right) \phi^{CDS}$

Functional form for η – Linked to Peclet number, Pe
Many ad-hoc choices, however, a common physical approach is tanh



$$Pe = \frac{\rho U L}{\mu}$$

$$\eta = \frac{1}{2}\left[1 + tanh\left(\frac{Pe - c_{trans}}{c_{width}}\right)\right]$$

- peclet_function_form:
    velocity: tanh
    mixture_fraction: tanh

- peclet_function_tanh_transition:
    velocity: 5000.0
    mixture_fraction: 2.0

- peclet_function_tanh_width:
    velocity: 200.0
    mixture_fraction: 4.0

Hybrid-Blending Sanity Check

Consider a simple fluids case where we have air (298.15K) flowing 1 m/s in a 1 m^3 domain

- For Pe = 2 at each element, we would require ~0.0305 m resolution, or a mesh of size: ~35,200

Consider a simple fluids case where we have air (298.15K) flowing 10 m/s in a 1 m^3 domain

- For Pe = 2 at each element, we would require ~0.00305 m resolution, or a mesh of size: ~35,200,000

- In general, this constraint results in extremely high mesh counts for most all practical flow configurations that even approach a turbulent flow regime

- Moreover, we are blending with upwind – an already shown to be overly diffuse and non-energy conserving

Monotonic Issues at high Pe: Simple Matrix Analysis

Consider our passive scalar concentration whose natural range (as a mass fraction) is bounded between 0 and unity, here, as a stationary transport equation:

$$\frac{\partial \rho u_j \phi}{\partial x_j} - \frac{\partial}{\partial x_j}\left(\rho D \frac{\partial \phi}{\partial x_j}\right) = 0$$

Using our CDS and standard diffusion operator yields the following matrix system:

$$\left(\frac{\rho u}{2}\begin{bmatrix}-1 & 0 & 1\end{bmatrix} + \frac{\rho D}{\Delta x}\begin{bmatrix}-1 & 2 & -1\end{bmatrix}\right)\begin{bmatrix}\phi_{i-1}\\ \phi_i \\ \phi_{i+1}\end{bmatrix}.$$

$$a_{i,i-1} = \frac{\rho D}{\Delta x}\left(1 + \frac{Pe}{2}\right)$$

With coefficients, $a_{i,i}\phi_i = a_{i,i-1}\phi_{i-1} + a_{i,i+1}\phi_{i+1}$ and: $a_{i,i} = (a_{i,i-1} + a_{i,i+1})$

$$a_{i,i+1} = \frac{\rho D}{\Delta x}\left(1 - \frac{Pe}{2}\right)$$

Positive for Pe < 2

Substituting: $a_{i,i-1} = a_{i,i} - a_{i,i+1}$ and defining: $\xi = \frac{a_{i,i+1}}{a_{i,i}}$

Yields: $\phi_i = \xi\phi_{i+1} + (1-\xi)\phi_{i-1}$

For Pe < 2, the value of the scalar at node i is <u>a linear combination of the neighboring values</u>

Monotonic Issues at high Pe: Alternative View (Diagonal Dominance)

Define Diagonal Dominance as: $\frac{\sum_{i \neq j} |a_{i,j}|}{|a_{i,i}|} \leq 1$

For a monotonic operator and ease of solving the linear system, diagonal dominance is desired

In our advection/diffusion case, this is expressed as:

$$\frac{\left|\frac{\rho D}{\Delta x}\left(1 - \frac{Pe}{2}\right)\right| + \left|\frac{\rho D}{\Delta x}\left(1 + \frac{Pe}{2}\right)\right|}{\frac{2\rho D}{\Delta x}} \leq 1$$

Which, again, is only ensured when the Peclet number is less than two

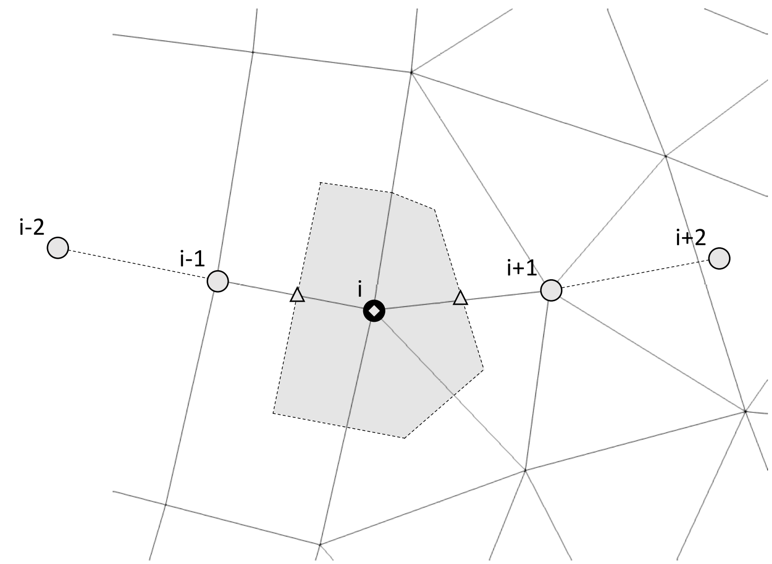
Alternatives to First-order Upwind: Higher-order Upwind – But how on an unstructured mesh?

Recall, that in the finite difference context, we could increase the upwind stencil to increase accuracy, e.g.,

Derivative	Accuracy	−8	−7	−6	−5	−4	−3	−2	−1	0
1	1								−1	1
	2							1/2	−2	3/2
	3						−1/3	3/2	−3	11/6

- However, for an unstructured setting, this approach is not easily achieved due to the local stencil connectivity
  - Recall, we are looping edges (as shown), or elements

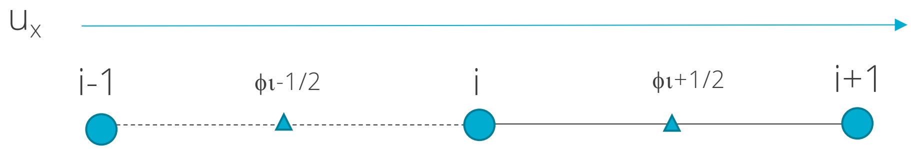
- Hint... Can we look towards the projected nodal gradient to effectively increase the stencil?



Classic Flux Limiters

Van Leer, B. (1974), "Towards the ultimate conservative difference scheme II. Monotonicity and conservation combined in a second order scheme", J. Comput. Phys., 14 (4): 361–370

- Consider our standard three-point stencil obtained by iterating edge 1 and 2



- In the above stencil, we are stressing that when iterating edge 2, we do not have information easily obtain for edge 1 (shown above as a dashed line); (speaking from an unstructured perspective)

$$\phi_{i+\frac{1}{2}} = \phi_{i+\frac{1}{2}}^{LOW} - \Phi(r_{i+\frac{1}{2}}) \left(\phi_{i+\frac{1}{2}}^{LOW} - \phi_{i+\frac{1}{2}}^{HIGH} \right)$$

- Above, the “LOW” and ”HIGH” are any operators that you select, e.g.,

$$\phi_{i+\frac{1}{2}}^{LOW} = \phi_i \qquad \phi_{i+\frac{1}{2}}^{HIGH} = \frac{\phi_i + \phi_{i+1}}{2} \longrightarrow \phi_{i+\frac{1}{2}} = \phi_i + \frac{1}{2}\Phi(r_{i+\frac{1}{2}}) \left(\phi_{i+1} - \phi_i \right)$$

$$r_{i+\frac{1}{2}} = \frac{(\phi_i - \phi_{i-1})}{(\phi_{i+1} - \phi_i)} \qquad (\phi_i - \phi_{i-1}) = 2G_x\phi_i\Delta x - (\phi_{i+1} - \phi_i) \qquad G_x\phi_i = \frac{\phi_{i+1} - \phi_{i-1}}{2\Delta x}$$

Flux Limiter Definition

Sweby (1984) defined set of permissible limiter regions for the desired second-order accurate methods: A sampling of the Sweby Diagram

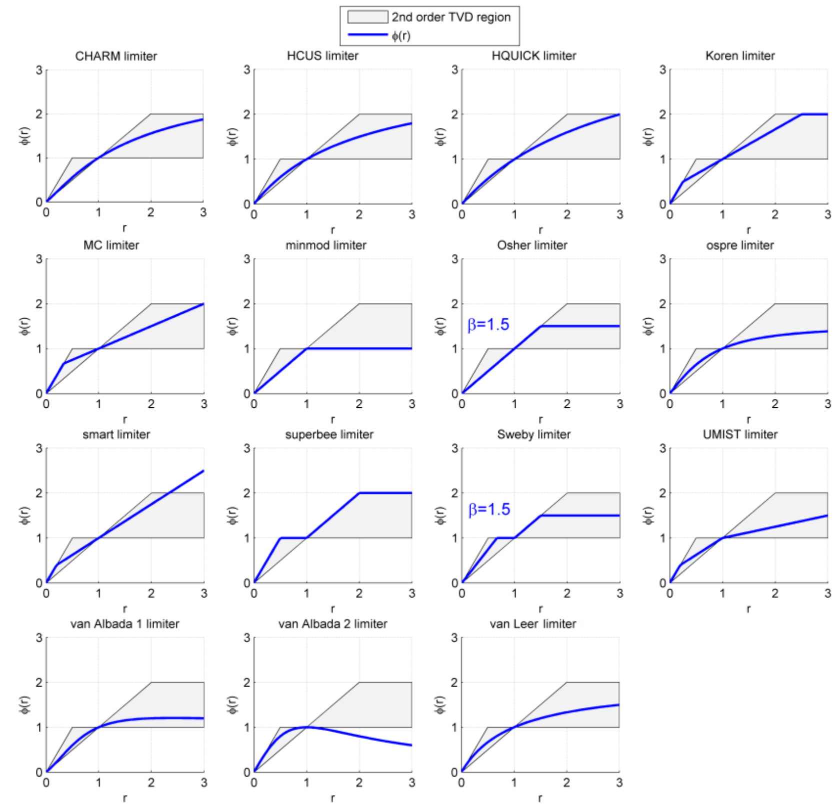
Superbee: $$\Phi(r) = max[0, min[2r, 1], min[r/2]]$$

Van Leer: $$\Phi(r) = \frac{r + |r|}{1 + |r|}$$

Symmetry property: $$\Phi(1/r) = \frac{\Phi(r)}{r}$$

Total Variation: $$TV(\phi) = \sum_{j=1}^{N} |\phi_j - \phi_{j-1}|$$

- For a monotonically increasing function, TV(ϕ) = $|\phi_1 - \phi_N|$. Note that if ϕ_1 and ϕ_N are taken constant, then, as long as the function remains monotonic the total variation is constant
- However, if TV(ϕ) increases, then this suggest oscillations in the solution have occurred



Sweby's 2nd order TVD region. Created in Matlab. Griffgruff 18:37, 11 October 2006 (UTC)

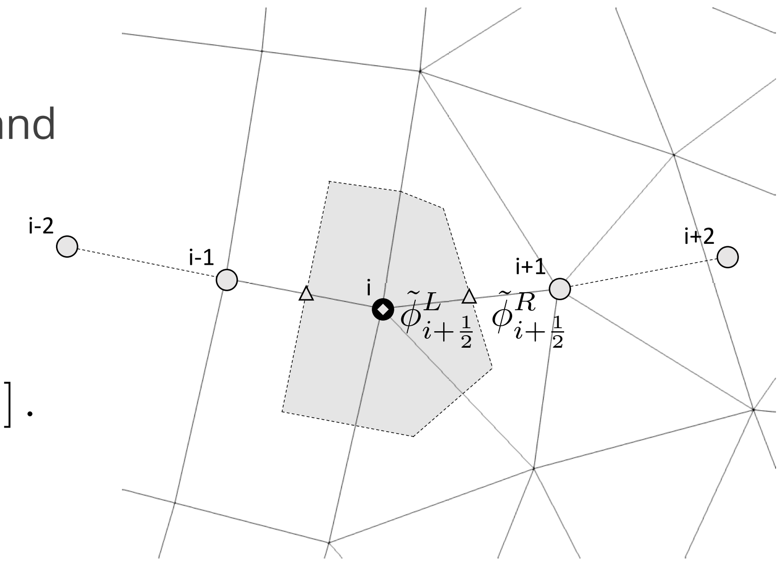
General Kappa-Method of Hirsh

Numerical Computation of Internal and External Flows, vol. 2, John Wiley & Sons, 1990

For the edge defined by the i and i+1 node, define a Left and Right state:

$$\tilde{\phi}^L_{i+\frac{1}{2}} = \phi_i + \frac{1}{4}\left[(1-\kappa)\left(\phi_i - \phi_{i-1}\right) + (1+\kappa)\left(\phi_{i+1} - \phi_i\right)\right],$$

$$\tilde{\phi}^R_{i+\frac{1}{2}} = \phi_{i+1} - \frac{1}{4}\left[(1+\kappa)\left(\phi_{i+1} - \phi_i\right) + (1-\kappa)\left(\phi_{i+2} - \phi_{i+1}\right)\right].$$



- For κ = +1, we simply revert to CDS
- For κ = -1, Second-order upwind
- For κ = 2/3, QUICK (Leonard, "A stable and accurate convective modelling procedure based on quadratic up-stream interpolation", Comput. Methods Appl. Mech. Eng. 19 (1979) 59–98.)

Assignment: Algebra!!!

Kappa = 0 Method of Hirsh

Numerical Computation of Internal and External Flows, vol. 2, John Wiley & Sons, 1990.

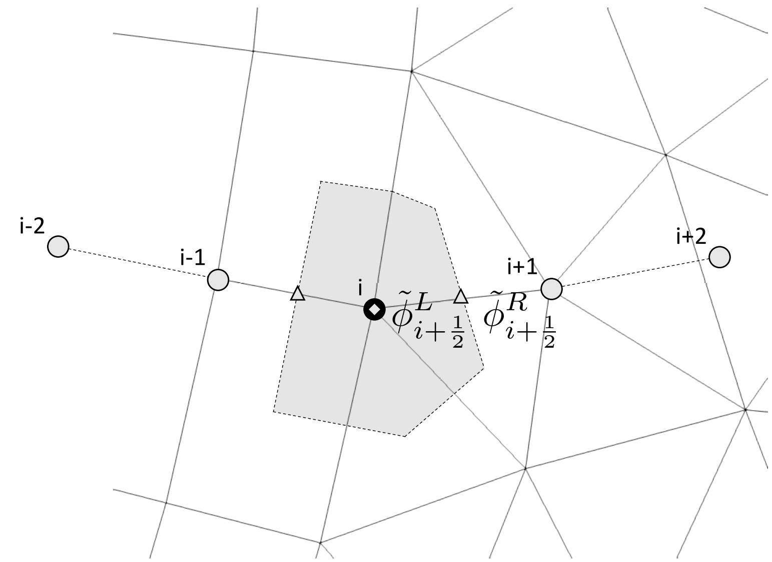
- For κ = 0, recast as:

$$\tilde{\phi}^L_{i+\frac{1}{2}} = \phi_i + \Phi^L \Delta x^L_j G_j \phi_i,$$
$$\tilde{\phi}^R_{i+\frac{1}{2}} = \phi_{i+1} - \Phi^R \Delta x^R_j G_j \phi^{i+1}$$

Where,

$$\Delta x^L_j = x^{ip}_j - x^L_j,$$
$$\Delta x^R_j = x^R_j - x^{ip}_j$$

- Above, define a "limiter" function Φ^L, Φ^R that "senses" when the solution is smooth (tends towards unity) and when the solution is oscillatory (tends towards zero)



Derived by substituting κ = 0, and using the projected nodal gradient definition – or – just by noting an extrapolation using a gradient

Assignment: Algebra!!!

Blending Approaches to Arrive at Pseudo Higher-order Methods: Upwind and Central

Define an upwind operator now at an arbitrary integration point:

$$\phi_{ip}^{UPW} = \alpha_{upw}\tilde{\phi}_{ip}^{L} + (1-\alpha_{upw})\,\phi_{ip}^{CDS}; \dot{m} > 0,$$
$$\alpha_{upw}\tilde{\phi}_{ip}^{R} + (1-\alpha_{upw})\,\phi_{ip}^{CDS}; \dot{m} < 0.$$

And for a generalized CDS scheme:

$$\phi_{ip}^{GCDS} = \frac{1}{2}\left(\hat{\phi}_{ip}^{L} + \hat{\phi}_{ip}^{R}\right), \qquad \begin{aligned} \hat{\phi}_{ip}^{L} &= \alpha\tilde{\phi}_{ip}^{L} + (1-\alpha)\,\phi_{ip}^{CDS}, \\ \hat{\phi}_{ip}^{R} &= \alpha\tilde{\phi}_{ip}^{R} + (1-\alpha)\,\phi_{ip}^{CDS} \end{aligned}$$

The Idealized Stencil Set

With Nalu input file specifications

$i-2$	$i-1$	i	$i+1$	$i+2$	α	α_{upw}
0	$-\frac{1}{2}$	0	$+\frac{1}{2}$	0	0	n/a
$+\frac{1}{8}$	$-\frac{6}{8}$	0	$+\frac{6}{8}$	$-\frac{1}{8}$	$\frac{1}{2}$	n/a
$+\frac{1}{12}$	$-\frac{8}{12}$	0	$+\frac{8}{12}$	$-\frac{1}{12}$	$\frac{2}{3}$	n/a
$+\frac{1}{4}$	$-\frac{5}{4}$	$+\frac{3}{4}$	$+\frac{1}{4}$	0	$\dot{m} > 0$	1
0	$-\frac{1}{4}$	$-\frac{3}{4}$	$+\frac{5}{4}$	$-\frac{1}{4}$	$\dot{m} < 0$	1
$+\frac{1}{6}$	$-\frac{6}{6}$	$+\frac{3}{6}$	$+\frac{2}{6}$	0	$\dot{m} > 0$	$\frac{1}{2}$
0	$-\frac{2}{6}$	$-\frac{3}{6}$	$+\frac{6}{6}$	$-\frac{1}{6}$	$\dot{m} < 0$	$\frac{1}{2}$

- alpha_upw:
  velocity: 1.0

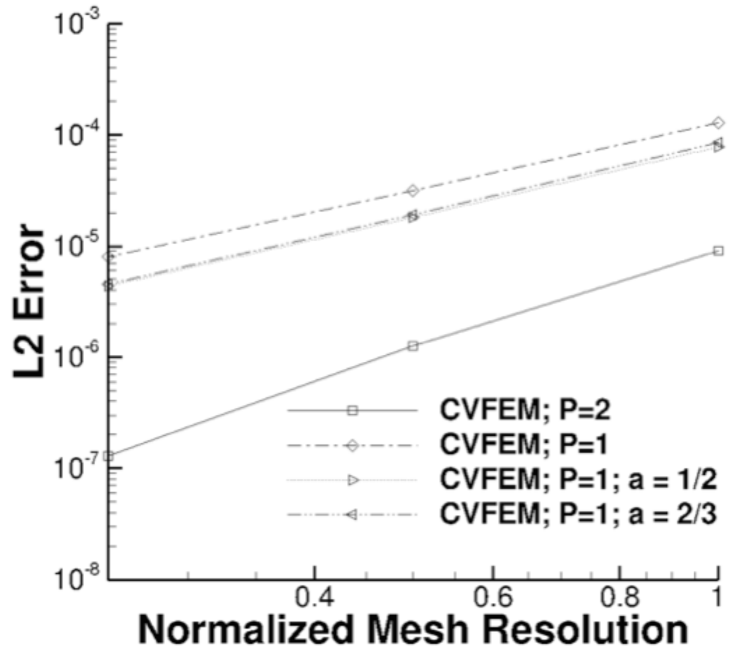
- alpha:
  velocity: 1.0

- upw_factor:
  velocity: 1.0 (zero reverts to first-order)

- limiter:
  velocity: yes

Pseudo 4th order Verification Results

Verification using Central (linear and quadratic) compared to pseudo 4th order



Lower error, however, still formally second-order accurate